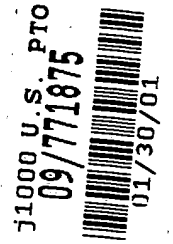


#2

Docket No. 1081.1107/JDH

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:)	
)	
Yoshitomo KUMAGAI)	
)	Group Art Unit: Unassigned
Serial No.: To be assigned)	
)	Examiner: Unassigned
Filed: January 30, 2001)	
)	
For: APPLICATION TRANSFER)	
METHOD, APPLICATION)	
TRANSFER SYSTEM AND)	
STORAGE MEDIUM)	
THEREFORE)	



**SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN
APPLICATION IN ACCORDANCE
WITH THE REQUIREMENTS OF 37 C.F.R. §1.55**

*Assistant Commissioner for Patents
Washington, D.C. 20231*

Sir:

In accordance with the provisions of 37 C.F.R. §1.55, the applicant submits herewith a certified copy of the following foreign application:

Japanese Patent Application No. 2000-259225
Filed: August 29, 2000.

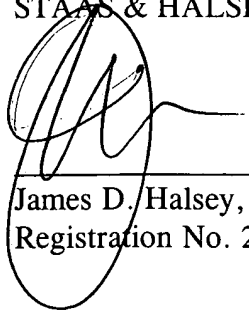
It is respectfully requested that the applicant be given the benefit of the foreign filing date as evidenced by the certified papers attached hereto, in accordance with the requirements of 35 U.S.C. §119.

Respectfully submitted,

STAAS & HALSEY LLP

Date: January 30, 2001

By: _____


James D. Halsey, Jr.
Registration No. 22,729

700 Eleventh Street, N.W., Suite 500
Washington, D.C. 20001
(202) 434-1500

日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日

Date of Application:

2000年 8月29日

出 願 番 号

Application Number:

特願2000-259225

出 願 人

Applicant (s):

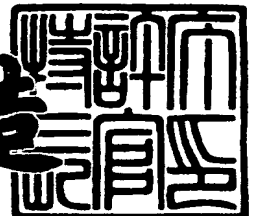
富士通株式会社



2000年11月17日

特許庁長官
Commissioner,
Patent Office

及川耕造



出証番号 出証特2000-3096818

【書類名】 特許願

【整理番号】 0051177

【提出日】 平成12年 8月29日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 3/14

【発明の名称】 アプリケーションの移植方法及びそのシステム

【請求項の数】 5

【発明者】

【住所又は居所】 神奈川県横浜市港北区新横浜二丁目 1 5 番 1 6 株式
会社富士通コンピュータテクノロジー内

【氏名】 熊谷 善友

【特許出願人】

【識別番号】 000005223

【氏名又は名称】 富士通株式会社

【代理人】

【識別番号】 100094514

【弁理士】

【氏名又は名称】 林 恒▲徳▼

【代理人】

【識別番号】 100094525

【弁理士】

【氏名又は名称】 土井 健二

【手数料の表示】

【予納台帳番号】 030708

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9704944

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 アプリケーションの移植方法及びそのシステム

【特許請求の範囲】

【請求項 1】 移植元の環境におけるアプリケーションを、移植先の環境におけるアプリケーションに移植する方法において、

前記移植元の環境において、前記移植元のアプリケーションの G U I 定義ファイルを使用して、メニューを表示して状態を確認するステップと、

前記表示された状態から得られる G U I 情報を追加した前記移植先の環境におけるアプリケーションの G U I 定義ファイルを作成するステップとを有することを

特徴とするアプリケーションの移植方法。

【請求項 2】 前記移植元の環境におけるアプリケーションのインタフェイス層を、前記作成された G U I 定義ファイルを読み込むように、書き換え、前記移植先の環境におけるアプリケーションに移植するステップを更に有することを

特徴とする請求項 1 のアプリケーションの移植方法。

【請求項 3】 前記作成ステップは、

前記移植元のアプリケーションの G U I 定義ファイルを使用して、作成したメニューを表示しうる前記移植先の環境におけるアプリケーションの G U I 定義ファイルを作成するステップからなることを

特徴とする請求項 1 のアプリケーションの移植方法。

【請求項 4】 前記確認ステップは、

前記メニューの親ウィンドウから順次子ウィンドウを辿っていき、それぞれのウィンドウの表示状態における位置とサイズを取り出すステップからなり、

前記作成ステップは、

前記取り出した位置とサイズを出力して、前記 G U I 定義ファイルを作成するステップからなることを

特徴とする請求項 1 のアプリケーション移植方法。

【請求項 5】 移植元の環境におけるアプリケーションを、移植先の環境におけるアプリケーションに移植するシステムにおいて、

前記移植元のアプリケーションのGUI定義ファイルと、
表示装置と、

前記移植元の環境において、前記移植元のアプリケーションのGUI定義ファイルを使用して、メニューを表示して状態を確認し、前記表示された状態から得られるGUI情報を追加した前記移植先の環境におけるアプリケーションのGUI定義ファイルを作成する作成部とを有することを

特徴とするアプリケーションの移植システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、1のプラットフォームを使用して作成したアプリケーションプログラムを他のプラットフォームに移植するアプリケーションの移植方法及びそのシステムに関し、特に、アプリケーションのGUI（グラフィカルユーザーインターフェイス）のメニューデータを利用できるようにしたアプリケーションの移植方法及びそのシステムに関する。

【0002】

【従来の技術】

図11は、従来技術を説明するためのアプリケーションの構造図である。1のOS（オペレーティングシステム）のアプリケーションプログラム（以下、アプリケーションという）は、そのまま他のOS環境で使用できない。このアプリケーションを他のOS環境で使用するため、他の環境に移植する必要がある。以下、ワークステーション（WS）のUNIXアプリケーションを、パソコン（PC）のWindowsアプリケーションに移植する例で説明する。

【0003】

例えば、UNIX上で、X11/Motif（第1のプラットフォーム）120を使用して作成したUNIXアプリケーションプログラム100は、X11/Motif依存部分110を有する。一方、WindowsNTアプリケーションプログラム200は、WindowsNT依存部分210を有する。

【0004】

このUNIXアプリケーション100を、Windows NTに移植するには、次のような方法が利用されていた。

(1) アプリケーション100のX11/Motif依存部分110を、全てWindows NTに依存する機能に書き換える。

(2) アプリケーション100はそのまま、X11/Motif依存部分110を、Windows NTに依存する部分が動作するようにエミュレーションするツール(プログラム)を利用する。

【0005】

しかし、前者は、個々のアプリケーション毎に、書き換える必要があり、移植するアプリケーションが複数ある場合は、書き換え量が大きくなる。又、後者は、各利用者が、エミュレーションツールを購入する必要がある他に、エミュレーションツールをアプリケーション実行時に常時起動しておく必要がある。更に、エミュレーションツールに障害があった場合には、エミュレートできないか、エミュレートに時間がかかる。

【0006】

このため、アプリケーションとプラットフォームとの間を取り持つインタフェイス層を設け、アプリケーションは、そのインタフェイスを使用して作成することにより、プラットフォームが変わっても、インタフェイス部分を書き換えるだけで、アプリケーションの移植を行う方法が提案されている。

【0007】

【発明が解決しようとしている課題】

近年のアプリケーションは、ユーザーが利用しやすいように、GUI(グラフィカルユーザーインタフェイス)を持つ。GUIは、GUI定義により、所望のメニューを表示する。GUI定義の内容は、使用しているGUIツールにより、その内容が異なるため、移植元と移植先で使用するGUIツールが異なる場合には、メニューを作成する時に、必要となる情報に過不足が生じ、移植先で正しいメニューの作成が困難となるという問題がある。

【0008】

例えば、UNIX上で、X11/Motif(第1のプラットフォーム)12

0を使用して作成したUNIXアプリケーションプログラム100を、Windows NT（第2のプラットフォーム）に移植する場合には、X11/MotifのGUI定義では、メニューの外枠の位置とサイズが定義されているだけであり、メニューを構成する各部品1つ1つの位置とサイズは定義されていない。しかし、Windows NTでは、メニューの中に配置された各部品を作成する場合は、部品の位置と大きさが必要となる。

【0009】

部品の位置とサイズ以外に、GUIを定義する情報としては、色、フォント名、描画方向、配列、陰影、ピクスマップ名、部品の親子関係（階層関係）、部品の配置方法とサイズ変更時の振る舞い、部品の状態（On/Off）、スケール、テキスト部品等の部品自身が持つ値、文字列等がある。

【0010】

従来技術では、移植先でGUI定義情報が不足する場合には、人手で入力する必要があり、その作業には、多大な工数を必要とするという問題があった。

【0011】

従って、本発明の目的は、移植元で作成されたアプリケーションプログラムのGUI定義を移植先で利用して、同一メニューを生成するためのアプリケーションの移植方法及びそのシステムを提供するにある。

【0012】

又、本発明の他の目的は、少ない工数でアプリケーションプログラムを移植するためのアプリケーションの移植方法及びそのシステムを提供するにある。

【0013】

【課題を解決するための手段】

この目的の達成のため、本発明のアプリケーション移植方法は、移植元の環境において、移植元のアプリケーションのGUI定義ファイルを使用して、メニューを表示して状態を確認するステップと、表示された状態から得られるGUI情報を追加した前記移植先の環境におけるアプリケーションのGUI定義ファイルを作成するステップとを有する。

【0014】

本発明では、移植元のアプリケーションのGUI定義ファイルを使用して、メニューを表示し、表示された状態から得られるGUI情報を追加した移植先の環境におけるアプリケーションのGUI定義ファイルを作成するため、移植元のアプリケーションのGUI定義から移植先アプリケーションのGUI定義を自動作成することができる。

【0015】

又、本発明では、好ましくは、移植元の環境におけるアプリケーションのインタフェース層を、前記作成されたGUI定義ファイルを読み込むように、書き換え、前記移植先の環境におけるアプリケーションに移植するステップを更に有することにより、容易に移植先アプリケーションで作成されたGUI定義を使用して、メニューを表示できる。

【0016】

更に、本発明では、好ましくは、前記作成ステップは、前記移植元のアプリケーションのGUI定義ファイルを使用して、作成したメニューを表示しうる前記移植先の環境におけるアプリケーションのGUI定義ファイルを作成するステップからなることにより、メニュー内容の全体を移植できる。

【0017】

更に、本発明では、好ましくは、前記確認ステップは、前記メニューの親ウィンドウから順次子ウィンドウを辿っていき、それぞれのウィンドウの表示状態における位置とサイズを取り出すステップからなり、前記作成ステップは、前記取り出した位置とサイズを出力して、前記GUI定義ファイルを作成するステップからなることにより、ウィンドウ階層を利用して、少ない工数でGUI定義を自動移植できる。

【0018】

【発明の実施の形態】

図1は、本発明の一実施の形態のアプリケーション移植方法の説明図、図2は、図1のインタフェース層の構成図、図3及び図4は、図1のインタフェース層の説明図である。

【0019】

図 1 は、UNIX アプリケーション 1 0 - 1 ~ 1 0 - n を、Windows NT に移植する例を示す。図 1 に示すように、UNIX 上で、X 1 1 / M o t i f (第 1 のプラットフォーム) 1 2 を使用して作成した UNIX アプリケーション プログラム 1 0 - 1 ~ 1 0 - n の X 1 1 / M o t i f 依存部分を、複数のアプリケーションから共通に使用できるように、部品化 (ライブラリ) されたインタフェイス層 1 1 で構成した。

【 0 0 2 0 】

そして、UNIX アプリケーション 1 0 を Windows NT に移植するには、X 1 1 / M o t i f 依存部分のインタフェイス層 1 1 を、Windows NT のインタフェイス層 1 3 に作り変える。このインタフェイス層 1 3 を、プラットフォーム非依存部と、プラットフォーム依存部に分け、プラットフォーム依存部は、Windows NT 依存機能を使用し、プラットフォーム非依存部は、Windows NT のため、書き換える。

【 0 0 2 1 】

図 2 は、書き換えられた Windows NT アプリケーション プログラム 1 0 のインタフェイス層 1 3 の構成図である。図 2 に示すように、インタフェイス層 1 3 は、Windows NT 依存のプラットフォーム依存部 1 5 と、Windows NT のため作成されたプラットフォーム非依存部 1 6 からなる。

【 0 0 2 2 】

プラットフォーム依存部分 1 5 は、図形を描画する図形描画部 2 0 と、メニューを作成するメニュー作成部 2 1 と、キーボード、マウス 3 2 からのイベントを読み込む、イベント読込部 2 2 とからなり、Windows NT 依存機能を利用する。

【 0 0 2 3 】

プラットフォーム非依存部分 1 6 は、GUI 定義ファイル 3 4 から GUI 情報を読み込む GUI 管理部 2 4 と、イベント読込部 2 2 のイベントを解析するイベント管理部 2 7 と、アプリケーション本体 1 0、イベント管理部 2 7 の指定するメニューを解析し、GUI 情報を読み込むメニュー管理部 2 5 と、アプリケーション本体 1 0、イベント管理部 2 7 の指定する図形を解析する図形管理部 2 6 と

からなり、Windows NTに依存しない。

【0024】

図3及び図4は、図1及び図2のインターフェイス層11、13の説明図である。図3は、アプリケーション10がリスト作成(`gtXCreateList`())を指示した時のメニュー作成時のインターフェイス層11、13の処理の説明図である。図3に示すように、アプリケーション10がリスト作成を指示した場合、UNIXアプリケーション10のインタフェイス層11では、`Motif`リスト作成関数を呼び出し(`XmCreateList`())、`Motif`リスト作成関数は、GUI定義ファイル35のリスト情報(詳細は、図9で後述する)から`Motif`リストを作成する。

【0025】

一方、アプリケーション10がリスト作成を指示した場合、Windows NTアプリケーション10のインタフェイス層13では、GUI定義ファイル34のGUI情報を読み出し、Windowsリスト作成関数を呼び出し(`CreateWindow(ListBox, サイズ)`)、フォントを指定(`SendMessage(WM#SETFONT)`)する。即ち、Windowsリスト作成関数に位置とサイズ、フォントを指定する。Windowsリスト関数は、位置、サイズ、フォントを利用してWindowsリストを作成する。

【0026】

このように、メニューの作成には、Windowsアプリケーションでは、GUI情報(位置、サイズ、フォント)が必要であり、且つインタフェイス層13に、これを読み込む機能が必要である。

【0027】

次に、図形描画を図4により、説明する。図4に示すように、アプリケーション10が円の作成を指示した場合、UNIXアプリケーション10のインタフェイス層11では、図形属性を設定し(`XChangeGC`())、円描画関数を呼び出し(`XDrawCircle`())、X11の円描画関数は、ウィンドウに円を描画する。

【0028】

一方、アプリケーション10が円の作成を指示した場合、WindowsNTアプリケーション10のインタフェイス層13では、ペンとブラシを設定し(CreatePen(), CreateBrush()), 円描画関数を呼び出し(Ellipse()), Windowの円描画関数は、ウィンドウに円を描画する。

【0029】

このように、インタフェイス層を書き換える。

【0030】

次に、前述のように、X11/MotifのGUI定義では、メニューの外枠の位置とサイズが定義されているだけであり、メニューを構成する各部品1つ1つの位置とサイズは定義されていない。しかし、WindowsNTでは、メニューの中に配置された各部品を作成する場合は、部品の位置と大きさが必要となる。本発明では、X11/MotifのGUI定義をそのまま用いて、WindowsNTの部品(メニューのリスト等)の位置、サイズ情報を追加する。

【0031】

図5は、本発明の一実施の形態のGUI変換ツールの構成図、図6は、図5の変換プログラムの処理フロー図、図7は、その変換動作説明図、図8は、変換メニュー例の説明図、図9は、X11/MotifでのGUI定義の説明図、図10は、これを変換したWindowsNTでのGUI定義の説明図である。

【0032】

図5に示すように、UNIX環境における入出力関係は、X11/MotifのGUI定義35を用いて、UNIXアプリケーション10が表示装置1に、メニュー、図形を表示する。WindowsNT環境における入出力関係は、GUI定義34を用いて、WindowsNTアプリケーション10が表示装置2に、メニュー、図形を表示する。

【0033】

GUI定義変換プログラム4は、UNIX環境において、X11/MotifのGUI定義35を用いて、X11/MotifのGUI定義に不足している部品(メニューのリスト等)の位置、サイズを追加し、GUI定義ファイル34を

作成する。

【 0 0 3 4 】

具体的には、G U I 定義 3 5 を読み込んで作成したメニューは、親ウィンドウを頂点にして、子ウィンドウが階層的に順次配置されているため、一番上の親ウィンドウから順次子ウィンドウを辿ることができる。そこで、全てのウィンドウを辿っていき、それぞれのウィンドウの表示状態における位置とサイズを取り出し、G U I 定義の記述形式に従って出力することにより、位置とサイズが入った G U I 定義ファイル 3 4 を生成する。

【 0 0 3 5 】

より具体的に、図 7 及び図 8 を用いて、図 6 により G U I 定義変換プログラム 4 を説明する。

【 0 0 3 6 】

(S 1 0) 表示装置 5 で変換指示された変換指示ファイルを読み込む。

【 0 0 3 7 】

(S 1 1) G U I 定義情報をファイル 3 5 から読み込む。

【 0 0 3 8 】

(S 1 2) 読み込んだ G U I 情報を使用して、指示されたメニューを作成して表示する。即ち、表示装置 5 の指示により、最上位のウィンドウを、読み込んだ G U I 情報を用いて作成し、表示する。表示装置 5 のマウスでサイズを調整する。この最上位ウィンドウの G U I 情報を格納する。リソース内の下位ウィンドウ指定情報を取り出す。下位ウィンドウ指定の有無を確認する。無い場合には、ステップ S 1 3 に進む。一方、有る場合には、下位ウィンドウを読み込んだ G U I 情報を用いて作成し、表示する。表示装置 5 のマウスでサイズを調整する。この下位ウィンドウの G U I 情報を格納する。これを繰り返す。

【 0 0 3 9 】

図 8 の X 1 1 / M o t i f のメニュー 6 0 は、図 7 に示すように、最上位ウィンドウ 5 0 から順次ウィンドウが最下位ウィンドウ 5 4 ~ 5 8 に階層化されている。従って、最上位ウィンドウ 5 0 から最下位ウィンドウ 5 8 まで辿って表示していくことにより、各ウィンドウの G U I 情報（メニュー）が取得できる。

【 0 0 4 0 】

(S 1 3) 次に、確認用メッセージダイアログを表示し、応答を待つ。

【 0 0 4 1 】

(S 1 4) メッセージダイアログからの応答を調べる。NGなら、ステップ S 1 6 に進む。

【 0 0 4 2 】

(S 1 5) メッセージダイアログがOKなら、取得したメニュー情報をGUI定義情報に変換する。即ち、メニューの最上位ウィンドウのGUI情報を取り出し、GUI定義ファイル34に書き込む。下位ウィンドウの有無を確認する。無い場合には、ステップS16に進む。一方、有る場合には、下位ウィンドウのGUI情報を取り出し、GUI定義ファイル34に書き込む。これを繰り返す。

【 0 0 4 3 】

(S 1 6) 作成したメニューを削除して、終了する。

【 0 0 4 4 】

図9は、X11MotifのGUI定義例であり、図10は、これから作成した位置とサイズを追加したGUI定義である。図9のX11/MotifのGUI定義では、メニューの外枠の位置とサイズ(x, y, width, height)が定義されているだけであり、メニューを構成する各部品(menu, rc6, txt6, list7, scale7)の1つ1つの位置とサイズは定義されていない。

【 0 0 4 5 】

図10のWindowsNTのGUI定義では、メニューの外枠の位置とサイズ(x, y, width, height)が定義されている他に、メニューを構成する各部品(menu, rc6, txt6, list7, scale7)の1つ1つの位置とサイズが定義されている。

【 0 0 4 6 】

このようにして、X11/MotifのGUI定義35を用いて、X11/MotifのGUI定義に不足している部品(メニューのリスト等)の位置、サイズを追加し、GUI定義ファイル34を作成することができる。

【 0 0 4 7 】

又、GUI 定義 3 5 を読み込んで作成したメニューが、親ウィンドウを頂点にして、子ウィンドウが階層的に順次配置されているため、一番上の親ウィンドウから順次子ウィンドウを辿り、それぞれのウィンドウの表示状態における位置とサイズを取り出し、GUI 定義の記述形式に従って出力することにより、位置とサイズが入った GUI 定義ファイル 3 4 を自動生成できる。

【 0 0 4 8 】

上述した実施の形態では、X11/Motif を使用した UNIX アプリケーションから、WIN32 API を使用した Windows アプリケーションへの移植を例に説明したが、GUI 定義情報を表示した状態から取得できる他の GUI ツール及び各種 OS の組み合わせの相互間での移植に適用できる。例えば、GUI ツールとして、X11/Motif、WIN32 API の他に、X/View (OpenWindow), SunView, Java 等にも適用でき、移植の対象となる OS として、UNIX (Solaris, FreeBSD, Linux), Windows (NT/95/98/2000), MacOS, OS/2 等にも適用できる。

【 0 0 4 9 】

以上、本発明を、実施の形態で説明したが、本発明の趣旨の範囲内において、種々の変形が可能であり、これらを本発明の技術的範囲から排除するものではない。

【 0 0 5 0 】

(付記 1) 移植元の環境におけるアプリケーションを、移植先の環境におけるアプリケーションに移植する方法において、前記移植元の環境において、前記移植元のアプリケーションの GUI 定義ファイルを使用して、メニューを表示して状態を確認するステップと、前記表示された状態から得られる GUI 情報を追加した前記移植先の環境におけるアプリケーションの GUI 定義ファイルを作成するステップとを有することを特徴とするアプリケーションの移植方法。(1)

(付記 2) 前記移植元の環境におけるアプリケーションのインタフェース層を、前記作成された GUI 定義ファイルを読み込むように、書き換え、前記移植先

の環境におけるアプリケーションに移植するステップを更に有することを特徴とする付記 1 のアプリケーションの移植方法。(2)

(付記 3) 前記作成ステップは、前記移植元のアプリケーションの GUI 定義ファイルを使用して、作成したメニューを表示しうる前記移植先の環境におけるアプリケーションの GUI 定義ファイルを作成するステップからなることを特徴とする付記 1 のアプリケーションの移植方法。(3)

(付記 4) 前記確認ステップは、前記メニューの親ウィンドウから順次子ウィンドウを辿っていき、それぞれのウィンドウの表示状態における位置とサイズを取り出すステップからなり、前記作成ステップは、前記取り出した位置とサイズを出力して、前記 GUI 定義ファイルを作成するステップからなることを特徴とする付記 1 のアプリケーション移植方法。(4)

(付記 5) 前記移植元の環境が UNIX OS であり、前記移植先の環境が Windows OS であることを特徴とする付記 1 のアプリケーション移植方法。

【 0 0 5 1 】

(付記 6) 移植元の環境におけるアプリケーションを、移植先の環境におけるアプリケーションに移植するシステムにおいて、前記移植元のアプリケーションの GUI 定義ファイルと、表示装置と、前記移植元の環境において、前記移植元のアプリケーションの GUI 定義ファイルを使用して、メニューを表示して状態を確認し、前記表示された状態から得られる GUI 情報を追加した前記移植先の環境におけるアプリケーションの GUI 定義ファイルを作成する作成部とを有することを特徴とするアプリケーションの移植システム。(5)

(付記 7) 作成部は、前記移植元の環境におけるアプリケーションのインタフェース層を、前記作成された GUI 定義ファイルを読み込むように、書き換え、前記移植先の環境におけるアプリケーションに移植することを特徴とする付記 6 のアプリケーションの移植システム。

【 0 0 5 2 】

(付記 8) 前記作成部は、前記移植元のアプリケーションの GUI 定義ファイルを使用して、作成したメニューを表示しうる前記移植先の環境におけるアプリケーションの GUI 定義ファイルを作成することを特徴とする付記 6 のアプリケ

ーション移植システム。

【 0 0 5 3 】

（付記 9）前記作成部は、前記メニューの親ウィンドウから順次子ウィンドウを辿っていき、それぞれのウィンドウの表示状態における位置とサイズを取り出し、前記取り出した位置とサイズを出力して、前記 G U I 定義ファイルを作成することを特徴とする付記 6 のアプリケーション移植システム。

【 0 0 5 4 】

（付記 1 0）前記移植元の環境が U N I X O S であり、前記移植先の環境が W i n d o w s O S であることを特徴とする付記 6 のアプリケーション移植システム。

【 0 0 5 5 】

【発明の効果】

移植元のアプリケーションの G U I 定義ファイルを使用して、メニューを表示し、表示された状態から得られる G U I 情報を追加した移植先の環境におけるアプリケーションの G U I 定義ファイルを作成するため、移植元のアプリケーションの G U I 定義から移植先アプリケーションの G U I 定義を自動作成することができる。

【図面の簡単な説明】

【図 1】

本発明の一実施の形態のアプリケーション移植方法の説明図である。

【図 2】

図 1 のインタフェイス層の構成図である。

【図 3】

メニュー指定時の図 2 のインタフェイス層の説明図である。

【図 4】

図形描画時の図 2 のインタフェイス層の説明図である。

【図 5】

図 2 の G U I 定義変換システムの構成図である。

【図 6】

図 5 の G U I 定義変換プログラムの処理フロー図である。

【図 7】

図 6 の G U I 定義の階層構造図である。

【図 8】

図 7 の G U I 定義のメニュー例の説明図である。

【図 9】

図 5 の X 1 1 / M o t i f での G U I 定義の説明図である。

【図 1 0】

図 9 の G U I 定義を変換した W i n d o w s N T の G U I 定義の説明図である。

【図 1 1】

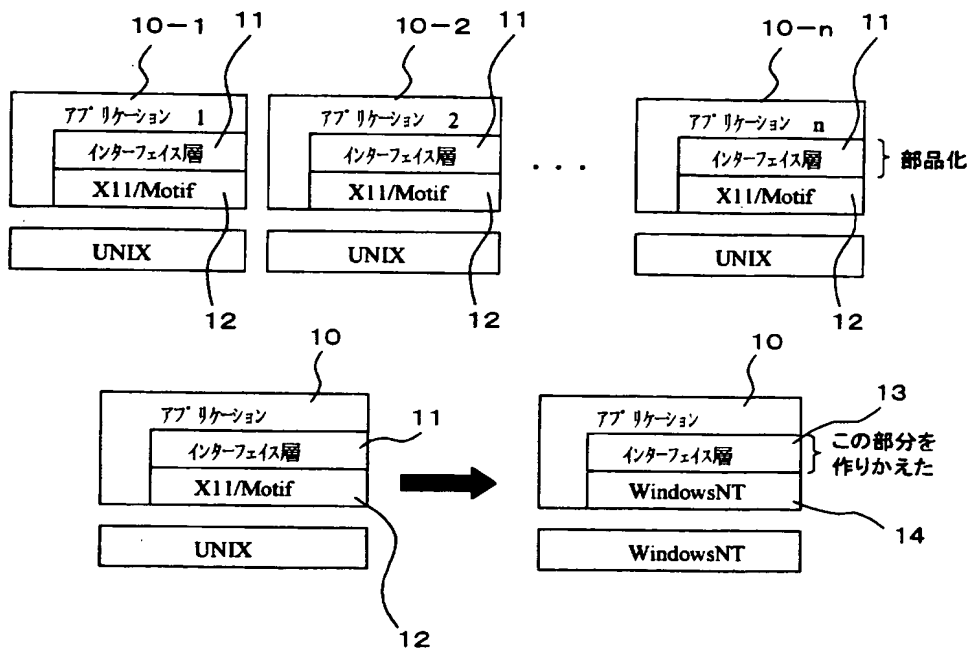
従来のアプリケーション移植方法の説明図である。

【符号の説明】

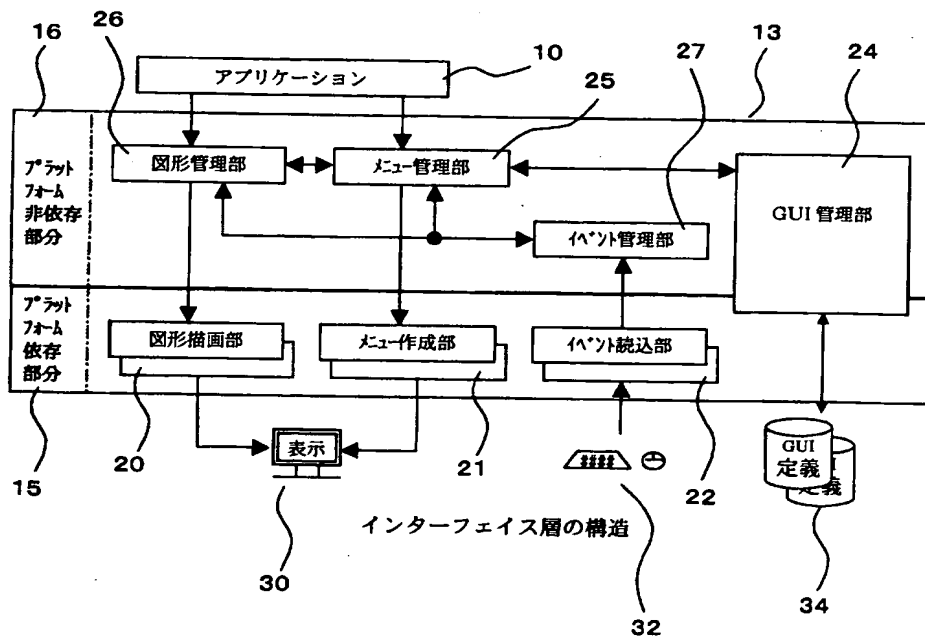
- 1、2、5 表示装置
- 4 G U I 定義変換プログラム
- 1 0 アプリケーションプログラム
- 1 1 インタフェイス層
- 1 2 第 1 のプラットフォーム
- 1 3 変換されたインタフェイス層
- 1 4 第 2 のプラットフォーム
- 1 5 プラットフォーム依存部分
- 1 6 プラットフォーム非依存部分
- 3 4 第 1 のプラットフォームの G U I 定義ファイル
- 3 5 第 2 のプラットフォームの G U I 定義ファイル

【書類名】 図面

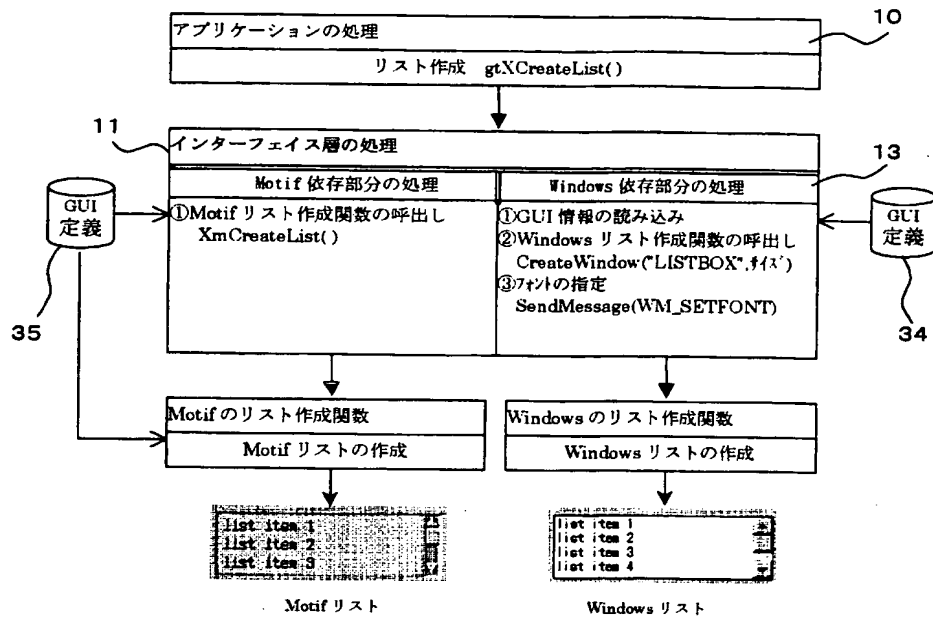
【図 1】



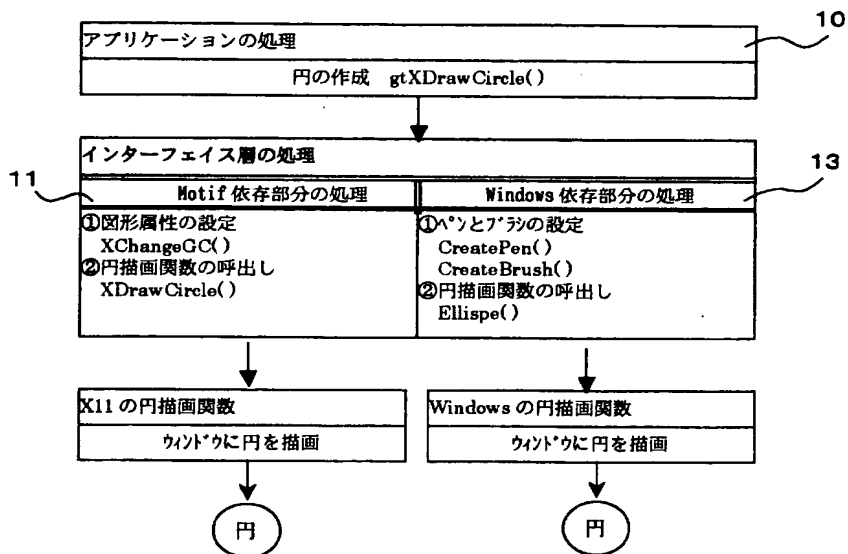
【図 2】



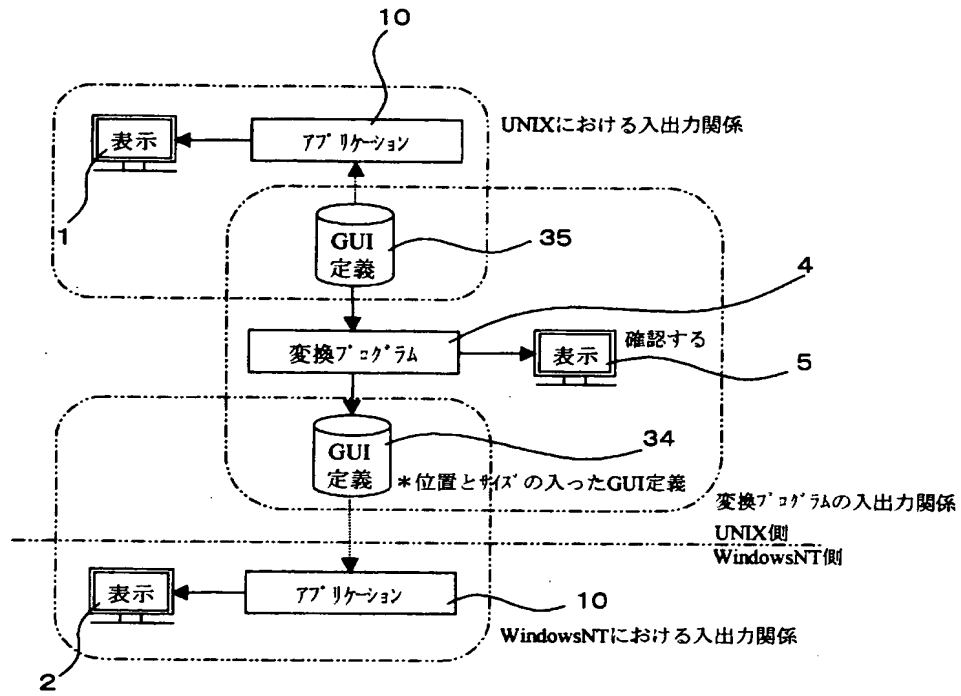
【図 3】



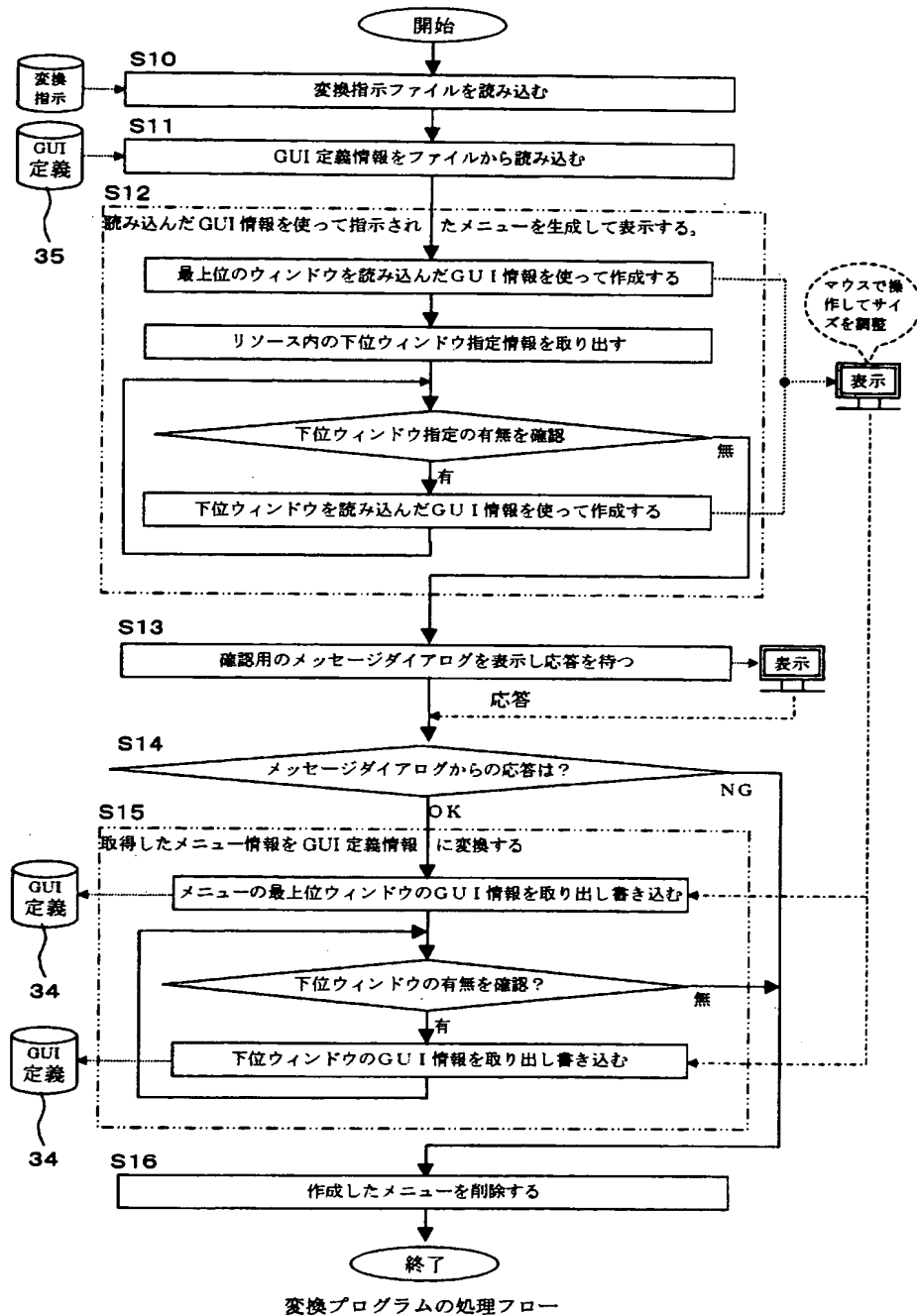
【図 4】



【図 5】

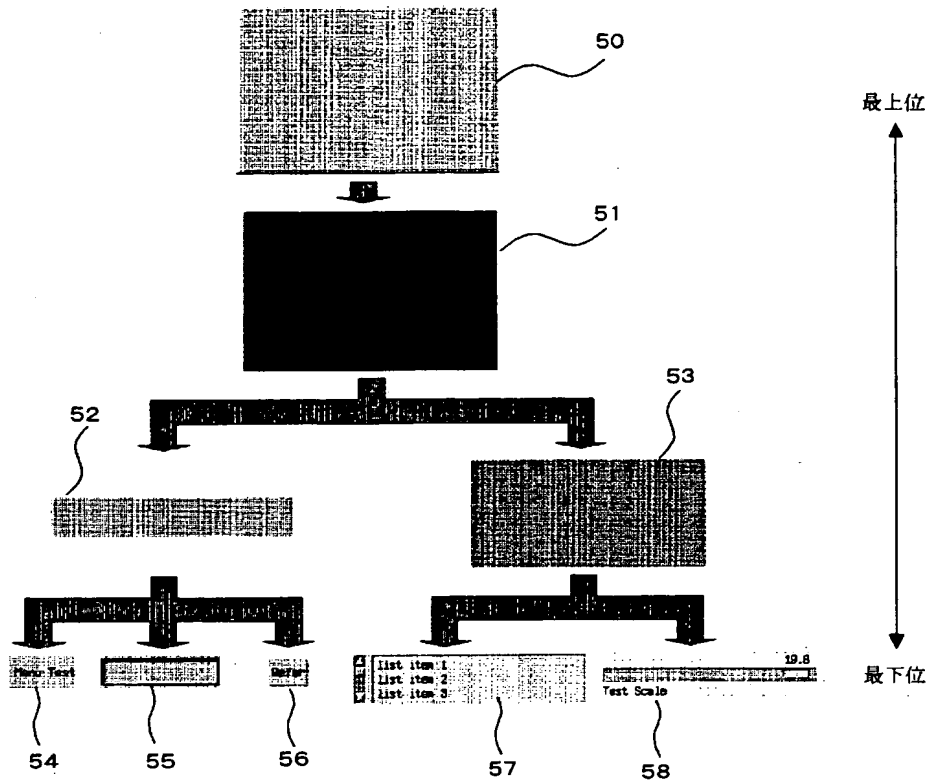


【図 6】



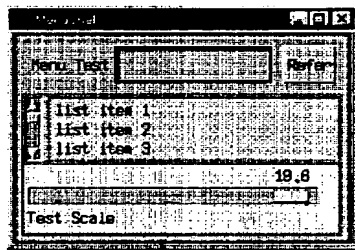
【図 7】

BEST AVAILABLE COPY

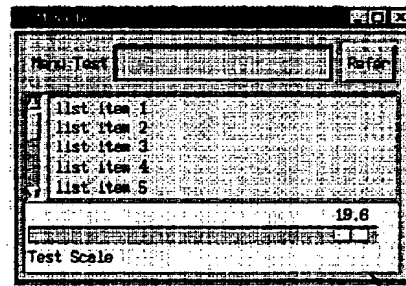


【図 8】

X11/Motif のメニュー例



WindowsNT のメニュー例



【図 9】

```

MenuShell.x: 100
MenuShell.y: 285
MenuShell.width: 308
MenuShell.height: 192
MenuShell*main.bottomOffset: 3
MenuShell*top.background: red
MenuShell*menu.rc6.orientation: HORIZONTAL
MenuShell*menu.rc6.background: cyan
MenuShell*menu.rc6.txt6.orientation: HORIZONTAL
MenuShell*menu.rc6.txt6.prefix.labelString: Menu Test
MenuShell*menu.rc6.txt6.prefix.y: 6
MenuShell*menu.rc6.txt6.text.prefix: True
MenuShell*menu.rc6.txt6.text.width: 160
MenuShell*menu.rc6.ref.labelString: Refer
MenuShell*menu.rc7.background: Green
MenuShell*menu.rc7.orientation: VERTICAL
MenuShell*menu.rc7*list7.background: pink
MenuShell*menu.rc7*list7.scrollBarDisplayPolicy: AS_NEEDED
MenuShell*menu.rc7*list7.listSizePolicy: CONSTANT
MenuShell*menu.rc7*list7.visibleItemCount: 3
MenuShell*menu.rc7*list7.items: list item 1, list item 2,
list item 3, list item 4, list item 5, list item 6
MenuShell*menu.rc7*list7.itemCount: 6
MenuShell*menu.rc7*list7.doubleClick: False
MenuShell*menu.rc7*list7.selectionPolicy: MULTIPLE_SELECT
MenuShell*menu.rc7*scale7.height: 60
MenuShell*menu.rc7*scale7.background: yellow
MenuShell*menu.rc7*scale7.decimalPoints: 1
MenuShell*menu.rc7*scale7.minimum: 0
MenuShell*menu.rc7*scale7.maximum: 200
MenuShell*menu.rc7*scale7.orientation: HORIZONTAL
MenuShell*menu.rc7*scale7.scaleHeight: 20
MenuShell*menu.rc7*scale7.scaleWidth: 280
MenuShell*menu.rc7*scale7.showValue: True
MenuShell*menu.rc7*scale7.titleString: Test Scale

```

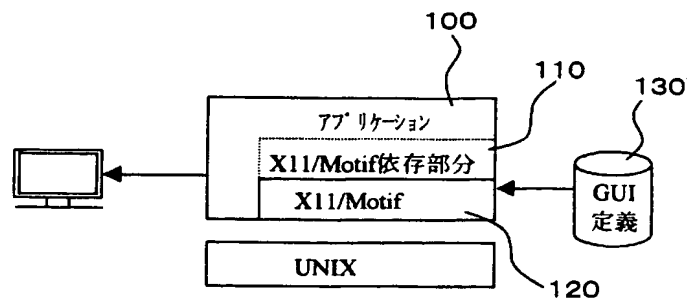
X11Motif の GUI 定義の例

【図 10】

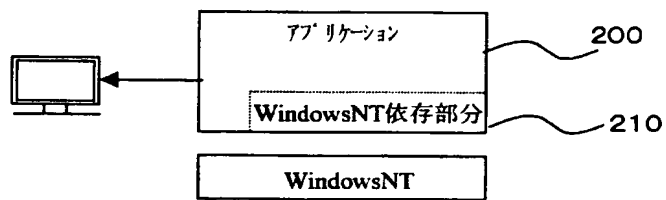
```
MenuShell.x: 100
MenuShell.y: 285
MenuShell.width: 308
MenuShell.height: 192
MenuShell*main.x: 3
MenuShell*main.y: 3
MenuShell*main.width: 298
MenuShell*main.height: 182
MenuShell*main*menu.x: 3
MenuShell*main*menu.y: 3
MenuShell*main*menu.width: 296
MenuShell*main*menu.height: 181
MenuShell*main*menu.rc6.x: 6
MenuShell*main*menu.rc6.y: 6
MenuShell*main*menu.rc6.width: 290
MenuShell*main*menu.rc6.height: 46
MenuShell*main*menu.rc6*txt6.x: 9
MenuShell*main*menu.rc6*txt6.y: 9
MenuShell*main*menu.rc6*txt6.width: 234
MenuShell*main*menu.rc6*txt6.height: 40
MenuShell*main*menu.rc6*txt6.prefix.x: 12
MenuShell*main*menu.rc6*txt6.prefix.y: 12
MenuShell*main*menu.rc6*txt6.prefix.width: 67
MenuShell*main*menu.rc6*txt6.prefix.height: 34
MenuShell*main*menu.rc6*txt6.text.x: 82
MenuShell*main*menu.rc6*txt6.text.y: 12
MenuShell*main*menu.rc6*txt6.text.width: 158
MenuShell*main*menu.rc6*txt6.text.height: 34
MenuShell*main*menu.rc6.ref.x: 246
MenuShell*main*menu.rc6.ref.y: 9
MenuShell*main*menu.rc6.ref.width: 47
MenuShell*main*menu.rc6.ref.height: 40
MenuShell*main*menu.rc7.x: 6
MenuShell*main*menu.rc7.y: 55
MenuShell*main*menu.rc7.width: 290
```

位置とサイズを追加した GUI 定義の例

【図 11】



X11/Motifで作ったUNIXアプリケーションの構造



WindowsNTアプリケーションの構造

【書類名】 要約書

【要約】

【課題】異なるプラットフォームへのアプリケーションの移植方法に関し、G U I 定義を利用して、移植先のG U I 定義を作成防止する。

【解決手段】移植元のアプリケーション（１）のG U I 定義ファイル（３５）を使用して、メニューを表示し、表示された状態から得られるG U I 情報を追加した移植先の環境におけるアプリケーションのG U I 定義ファイル（３４）を作成するため、移植元のアプリケーションのG U I 定義から移植先アプリケーションのG U I 定義を自動作成することができる。

【選択図】図５

認定・付加情報

特許出願の番号	特願 2000-259225
受付番号	50001096826
書類名	特許願
担当官	風戸 勝利 9083
作成日	平成 12 年 9 月 1 日

<認定情報・付加情報>

【特許出願人】

【識別番号】	000005223
【住所又は居所】	神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号
【氏名又は名称】	富士通株式会社

【代理人】

申請人

【識別番号】	100094514
【住所又は居所】	神奈川県横浜市港北区新横浜 3-9-5 第三東 昇ビル 3 階 林・土井 国際特許事務所
【氏名又は名称】	林 恒徳

【代理人】

【識別番号】	100094525
【住所又は居所】	神奈川県横浜市港北区新横浜 3-9-5 第三東 昇ビル 3 階 林・土井 国際特許事務所
【氏名又は名称】	土井 健二

出 願 人 履 歴 情 報

識別番号 [000005223]

1. 変更年月日	1996年 3月26日
[変更理由]	住所変更
住 所	神奈川県川崎市中原区上小田中4丁目1番1号
氏 名	富士通株式会社